IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

United States Provisional Patent Application

for

METHODS AND APPARATUS FOR IDENTIFYING RELATED NODES IN A DIRECTED GRAPH HAVING NAMED ARCS

Inventor:

Howard Greenblatt, a U.S. citizen residing at
22 Coolidge Road, Wayland, MA 01778

Alan Greenblatt, a U.S. citizen residing at
64 Hunt Road, Sudbury, MA 01776

David A. Bigwood, a U.S. citizen residing at
324 Concord Avenue, Lexington, MA 02421

Colin P. Britton, a U.S. citizen residing at
17 Pheasant Lane, Lexington, MA 02421

(Cover)

METHODS AND APPARATUS FOR IDENTIFYING RELATED NODES IN A DIRECTED GRAPH HAVING NAMED ARCS

Background of the Invention

The invention pertains to digital data processing and, more particularly, to methods and apparatus for identifying subsets of related data in a data set. The invention has application, for example, in enterprise business visibility and insight using real-time reporting tools.

It is not uncommon for a single company to have several database systems—separate systems not interfaced—to track internal and external planning and transaction data. Such systems might have been developed at different times throughout the history of the company and are therefore of differing generations of computer technology. For example, a marketing database system tracking customers may be ten years old, while an enterprise resource planning (ERP) system tracking inventory might be two or three years old. Integration between these systems is difficult at best, consuming specialized programming skill and constant maintenance expenses.

A major impediment to enterprise business visibility is the consolidation of these disparate legacy databases with one another and with newer databases. For instance, inventory on-hand data gleaned from a legacy ERP system may be difficult to combine with customer order data gleaned from web servers that support e-commerce (and other web-based) transactions. This is not to mention difficulties, for example, in consolidating resource scheduling data from the ERP system with the forecasting data from the marketing database system.

Even where data from disparate databases can be consolidated, e.g., through data mining, directed queries, brute-force conversion and combination, or otherwise, it may be difficult (if not impossible) to use. For example, the manager of a corporate marketing campaign may be wholly unable to identify relevant customers from a listing of tens, hundreds or even thousands of pages of consolidated corporate ERP, e-commerce, marketing and other data.

An object of this invention is to provide improved methods and apparatus for digital data processing and, more particularly, for identifying subsets of related data in a data set.

A related object is to provide such methods and apparatus as facilitate enterprise business visibility and insight.

A further object is to provide such methods and apparatus as can rapidly identify subsets of related data in a data set, e.g., in response to user directives or otherwise.

A further object of the invention is to provide such methods and apparatus as can be readily and inexpensively implemented.

Summary of the Invention

The foregoing are among the objects attained by the invention which provides, in one aspect, a method for identifying related data in a directed graph, such as an RDF data set. A "first" step—though the steps are not necessarily executed in sequential order—includes identifying (or marking) as related data expressly satisfying a criteria (e.g., specified by a user). A "second" step includes identifying as related ancestors of any data identified as related, e.g., in the first step, unless that ancestor conflicts with the criteria. A "third" step of the method is identifying descendents of any data identified, e.g., in the prior steps, unless that descendent conflicts with the criteria or has a certain relationship with the ancestor from which it descends. The methods generates, e.g., as output, an indication of each of the nodes identified as related in these steps.

By way of example, in the first step, a method according to this aspect of the invention can identify nodes in the directed graph that explicitly match a criteria in the form *field1* = *value1,* where *field1* is a characteristic (or attribute) of one or more of the nodes and *value1* is a value of the specific characteristic (or attribute). Of course, criteria are specific to the types of data in the data set and can be more complex, including for example, Boolean expressions and operators, wildcards, and so forth. Thus, for example, a criteria of a data set composed of RDF triples might be of the form *predicate=CTO* and *object=Colin*, which identifies, as related, triples having a predicate "CTO" and an object "Colin."

By way of further example, in second step, the method "walks" up the directed graph from each node identified as related in first step (or any of the steps) to find ancestor nodes. Each of these is identified as related unless it conflicts with the criteria. To continue the example, if the first step marks as related a first RDF triple that matches the criteria *predicate=CTO* and *object=Colin*, the second step marks as related a second, parent triple whose object is the subject of the first triple, unless that second (or parent) triple otherwise conflicts with the criteria, e.g., has another object specifying that Dave is the CTO.

By way of further example, in the third step, the method walks down the directed graph from each node identified in the previously described steps (or any of the steps) to find descendent nodes. Each of these is identified as related unless (i) it conflicts with the criteria or (ii) its relationship with the ancestor from which walking occurs is of the same type as the relationship that ancestor has with a child, if any, from which the ancestor was identified by operation of the second step. To continue the example, if the first step marks as related a first RDF triple that matches the criteria *predicate=CTO* and *object=Colin* and the second step marks as related a second, parent triple whose object is the subject of the first triple via a predicate relationship "Subsidiary," the third step marks as related a third, descendent triple whose subject is the object of the second, parent triple, unless that descendent triple conflicts with the criteria (e.g., has a predicate-object pair specifying that Dave is the CTO) or unless its relationship with the parent triple is also defined by a predicate relationship of type "Subsidiary."

As evident in the discussion above, according to some aspects of the invention, the data are defined by RDF triples and the nodes by subjects (or resource-type objects) of those triples. In other aspects, the data and nodes are of other data types—including, for example, meta directed graph data (of the type defined in one of the aforementioned incorporated-by-reference applications) where a node represents a plurality of subjects each sharing a named relationship with a plurality of objects represented by a node.

Still further aspects of the invention provide methods as described above in which the so-called first, second and third steps are executed in parallel, e.g., as by an expert system rule-engine. In other aspects, the steps are executed in series and/or iteratively.

In still further aspects of the invention, the invention provides methods for identifying related data in a directed graph by exercising only the first and second aforementioned steps. Other aspects provide such methods in which only the first and third such steps are exercised.

Still other aspects of the invention provide methods as described above in which the directed graph is made up of, at least in part, a data flow, e.g. of the type containing transactional or enterprise data. Related aspects provide such methods in which the steps are executed on a first portion of a directed graph and, then, separately on a second portion of the directed graph, e.g., as where the second portion reflects updates to a data set represented by the first portion.

These and other aspects are evident in the drawings and in the description that follows.

Brief Description of the Drawings

A more complete understanding of the invention may be attained by reference to the drawings, in which:

Figure 1 is a block diagram of a system according to the invention for identifying related data in a data set;

Figure 2 depicts a data set suitable for processing by a methods and apparatus according to the invention;

Figures 3–5 depict operation of the system of Figure 1 on the data set of Figure 2 with different criteria.

Detailed Description of the Illustrated Embodiment

Figure 1 depicts a system 8 according to the invention for identifying and/or generating (collectively, "identifying") a subset of a directed graph, namely, that subset matching or related to a criteria. The embodiment (and, more generally, the invention) is suited for use *inter alia* in generating subsets of RDF data sets consolidated from one or more data sources, e.g., in the manner described in the following copending, commonly assigned application, the teachings of which are incorporated herein by reference

United States Patent Application Number Serial No. 09/917,264, filed July 27, 2001, entitled "Methods and Apparatus for Enterprise Application Integration,"

United States Patent Application Number Serial No. 10/051,619, filed October 29, 2001, entitled "Methods And Apparatus For Real-time Business Visibility Using Persistent Schema-less Data Storage,"

United States Patent Application Number Serial No. 60/332,219, filed November 21, 2001, entitled "Methods And Apparatus For Calculation And Reduction Of Time-series Metrics From Event Streams Or Legacy Databases In A System For Real-time Business Visibility," and

United States Patent Application Number Serial No. 60/332,053, filed November 21, 2001, entitled "Methods And Apparatus For Querying A Relational Database Of RDF Triples In A System For Real-time Business Visibility."

The embodiment (and, again, more generally, the invention) is also suited *inter alia* for generating subsets of "meta" directed graphs of the type described in copending, commonly assigned application United States Patent Application Number Serial No. 10/138,725, filed May 3, 2002, entitled "Methods And Apparatus for Visualizing Relationships Among Triples of Resource Description Framework (RDF) Data Sets," the teachings of which are incorporated herein by reference.

The illustrated system 8 includes a module 12 that executes a set of rules 18 with respect to a set of facts 16 representing criteria in order to generate a subset 20 of a set of facts 10 representing an input data set, where that subset 20 represents those input data facts that match the criteria or are related thereto. For simplicity, in the discussion that follows the set of facts 16 representing criteria are referred to as "criteria" or "criteria 16," while the set of facts 10 representing data are referred to as "data" or "data 10." The illustrated system 8 is implemented on a general- or special-purpose digital data processing system, e.g., a workstation, server, mainframe or other digital data processing system of the type conventionally available in the marketplace, configured and operated in accord with the teachings herein. Though not shown in the drawing, the digital data processing system can be coupled for communication with other such devices, e.g., via a network or otherwise, and can include input/output devices, such as a keyboard, pointing device, display, printer and the like.

Illustrated module 12 is an executable program (compiled, interpreted or otherwise) embodying the rules 18 and operating in the manner described herein for identifying subsets of directed graphs. In the illustrated embodiment, module 12 is implemented in Jess (Java Expert System Shell), a rule-based expert system shell, commercially available from Sandia National Laboratories. However it can be implemented using any other "expert system" engine, if-then-else network, or other software, firmware and/or hardware environment (whether or not expert system-based) suitable for adaptation in accord with the teachings hereof.

The module 12 embodies the rules 18 in a network representation 14, e.g., an if-then-else network, or the like, native to the Jess environment. The network nodes are preferably executed so as to effect substantially parallel operation of the rules 18, though they can be executed so as to effect serial and/or iterative operation as well or in addition. In other embodiments, the rules are represented in accord with the specifics of the corresponding engine, if-then-else network, or other software, firmware and/or hardware environment on which the embodiment is implemented. These likewise preferably effect parallel execution of the rules 18, though they may effect serial or iterative execution instead or in addition.

The data set 10 is a directed graph, e.g., a collection of nodes representing data and directed arcs connecting nodes to one another. As used herein, a node at the source of an arc is referred to as an "ancestor" (or "direct ancestor"), while the node at the target of the arc is referred to herein as a "descendent" (or "direct descendent"). In the illustrated embodiment, each arc has an associated type or name, e.g., in the manner of predicates of RDF triples—which, themselves, constitute and/or form directed graphs.

By way of example, in addition to RDF triples, the data set 10 can comprise data structures representing a meta directed graph of the type disclosed in copending, commonly assigned United States Patent Application Serial No. 10/138,725, filed May 3, 2002, entitled "Methods And Apparatus for Visualizing Relationships Among Triples of Resource Description Framework (RDF) Data Sets, e.g., at Figure 4A - 6B and accompanying text, all of which incorporated herein by reference.

Alternatively or in addition, the data set 10 can comprise RDF triples of the type conventionally known in the art and described, for example, in Resource Description Framework

(RDF) Model and Syntax Specification (February 22, 1999). Briefly, RDF is a way of expressing the properties of items of data. Those items are referred to as *subjects* or *resources*. Their properties are referred to as *predicates*. And, the values of those properties are referred to as *objects*. In RDF, an expression of a property of an item is referred to as a *triple*, a convenience reflecting that the expression contains three parts: subject, predicate and object. Subjects can be anything that is described by an RDF expression. A predicate identifies a property of a subject. An object gives a "value" of a property. Objects can be *literals,* i.e., strings that identify or name the corresponding property (predicate). They can also be *resources*.

The data set 10 may be stored on disk for input to module 12. Alternatively, or in addition, the data set may be a data flow, e.g., a stream of data (real-time or otherwise) originating from e-commerce, point-of-sale or other transactions or sources (whether or not business- or enterprise-oriented). Moreover, the data set may comprise multiple parts, each operated on by module 12 at different times—for example, a first part representing a database and a second part representing updates to that database.

Criteria 16 contains expressions including, for example, literals, wildcards, Boolean operators and so forth, against which nodes in the data set are tested. In embodiments that operate on RDF data sets, the criteria can specify subject, predicate and/or object values or other attributes. In embodiments that operate on directed graphs of other types other appropriate values and attributes may be specified. Criteria can be input by a user, e.g., from a user interface, e.g., on an *ad hoc* basis. Alternatively or in addition, they can be stored and re-used, such as where numerous data sets exist of which the same criteria is applied. Further, the criteria 16 can be generated via dynamically, e.g., via other software (or hardware) applications.

Rules 18 define the tests for identifying data in the data set 20 that match the criteria or that are related thereto. These are expressed in terms of the types and values of the data items as well as their interrelationships or connectedness.

Rules applicable to a data set comprised of RDF triples can be expressed as follows:

Rule No.	Purpose	Rule
0 ("Criteria Rule")	Match criteria to triples in data set	If triple's object is a literal, identify triple as related if both triple's predicate and the object match those specified in the criteria. If triple's object is a resource, identify triple as related if triple's predicate matches that specified in criteria, if any, and if triples object matches that specified in criteria.
1 ("Sibling Rule")	Find as related other triples at the same level	Identify as related a triple that shares the same subject (i.e., siblings), except those siblings that have the same predicate as that specified in the criteria.
2 ("Ancestor Rule")	Walk up the directed graph to find valid triples.	Identify as related a triple that is a direct ancestor of a triple identified by any of the other rules and that is not in substantial conflict with the criteria; For purposes hereof, a triple whose object is the subject of another triple is deemed a direct ancestor of that other triple; a triple whose subject is the object of another triple is deemed a direct descendent of that other triple.

3 ("Descendent Rule")	Walk down the directed graph to find valid triples.	Identify as related a triple (hereinafter "identified descendent") that is a direct descendent of a triple (hereinafter "identified ancestor") identified as related by any of the other rules and which identified descendent (a) is not associated with the identified ancestor via a predicate substantially matching a predicate named in the criteria, if any, and (b) is not in substantial conflict with the criteria; (c) is not associated with the identified ancestor via a predicate matching a predicate by which the identified ancestor is associated with a triple, if any, as a result of which the identified ancestor was identified during execution of the Ancestor Rule.

As used above and throughout "substantial conflict" means conflict that is direct or otherwise material in regard to determining related data vis-a-vis the use for which the invention is employed (e.g., as determined by default in an embodiment and/or by selection made by a user thereof). By way of non-limiting example, for some uses (and/or embodiments) differences of any sort between the object of an RDF triple and that specified in a criteria are material, while for other uses (and/or embodiments) differences with respect to suffix, case and/or tense are immaterial. Those skilled in the art will appreciate that for other uses and/or embodiments, factors other than suffix, case and/or tense may be used in determining materiality or lack thereof.

Rules applicable to other directed graphs (e.g., not comprised of RDF triples) can be expressed as shown below. As noted above, these other directed graphs can include the aforementioned meta directed graphs, by way of non-limiting example. It will be appreciated that the rules which follow are functionally equivalent to those expressed above. However, they take into that the data nodes in those other directed graphs may have attributes in addition to those represented in their connectedness to other data nodes. To this end, the aforementioned Sibling Rule is subsumed in those aspects of the rules that follow which call for testing each data node to determine whether they conflict with the criteria.

Rule No.	**Purpose**	**Rule**
0 ("Criteria Rule")	Match criteria to data in data set	Identify as related data substantially matching a criteria;
1 (Ancestor Rule)	Walk up the directed graph to find valid data	Identified as related data that is a direct ancestor of data identified in any of these rules, and that is not in substantial conflict with the criteria;
2 (Descendent Rule)	Walk down the directed graph to find valid data	Identify as related data (hereinafter "identified descendent") that is a direct descendent of data (hereinafter "identified ancestor") identified as related in any of these rules, and which identified descendent: (a) Does not have a named relationship with the identified ancestor substantially matching a relationship named in the criteria, if any, and (b) Is not in substantial conflict with the criteria; and (c) Does not have a named relationship with the identified ancestor matching a relationship the identified ancestor has with a data, if any, as a result of which the identified ancestor was identified during execution of Rule 1.

Referring to back to Figure 1, the related data 20 output or otherwise generated by module 12 represents those nodes or triples identified as "related" during exercise of the rules. The data 20 can be output in the same form as the input data or some alternate form, e.g., pointers or other references to identified data within the data set 10. In some embodiments, it can be displayed via a user interface or printed, or digitally communicated to further applications for additional processing, e.g., via a network or the Internet. In one non-limiting example, the related data 20 can be used to generate mailings or to trigger message events.

In operation, the module 12 is loaded with rules 18. In the illustrated embodiment, this is accomplished via compilation of source code embodying those rules (expressed above in pseudo code) in the native or appropriate language of the expert system engine or other environment in which the module is implemented. *See,* step A. Of course, those skilled in the art will appreciate that, alternatively, rules in source code format can be retrieved at run time and interpreted instead of compiled.

The criteria 16 is then supplied to the module 12. *See,* step B. These can be entered by an operator, e.g., via a keyboard or other input device. Alternatively, or in addition, they can be retrieved from disk or input from another application (e.g., a messaging system) or device, e.g., via network, interprocess communication or otherwise.

The data set 10 is applied to the module 12 in step C. The data set 10 can be as described above, to wit, a RDF data set or other directed graph stored in a data base or contained in a data stream, or otherwise. The data set can be applied to the module 12 via conventional techniques known in the art, e.g., retrieval from disk, communication via network, or via any other technique capable of communicating a data set to a digital application.

In step D, the module 12 uses the rules 18 to apply the criteria 16 to the data set 10. In the illustrated embodiment, by way of non-limiting example, this step is executed via the network 14 configured (via the rules engine) in accord with the rules. In other embodiments, this step is executed via the corresponding internal representation of those rules.

Triples (in the case of RDF data sets) or data (in the case of data sets comprising other types of directed graphs) identified by the module as "related"—meaning, in the context hereof, that those triples match the criteria or are related thereto—are output as "identified data" in Step D. As described above, the output can be a list or other tabulation of identified data 20, or it can be a pointer or reference to that data, for example, a reference to a location within the data set 10.

In some embodiments, the output of identified data 20 can be stored for future use, e.g., for use with a mail-merge or other applications. In other embodiments, it can be digitally communicated to other data base systems or information repositories. Still further, in some embodiments, it can be added to a data base containing other related data, or even replace portions of that data based.

The table below lists a directed graph—here, the triples of an RDF data set—of the type suitable for processing by module 12 to identify data matching a criteria and related thereto. It will be appreciated that in practice, directed graphs processed by module 12 may contain hundreds, thousands or more nodes, e.g., as would be typical for an RDF set representing transactional and enterprise-related data. Moreover, it will be appreciated that the directed graphs and/or triples are typically expressed in a conventional data format (e.g., XML), or otherwise, for transfer to and from the module 12.

Subject	**Predicate**	**Object**
company://id#3	customer	company://id#1
company://id#3	customer	company://id#4
company://id#3	customer	company://id#2
company://id#1	employee	Howard
company://id#1	employee	Alan
company://id#1	CTO	Colin
company://id#2	employee	David
company://id#2	CTO	Colin

Figure 2 is a graphical depiction of this directed graph, i.e., RDF data set. Per convention, subjects and resource-type objects are depicted as oval-shaped nodes; literal-type objects are depicted as rectangular nodes; and predicates are depicted as arcs connecting those nodes.

Figure 3 depicts application by module 12 of criteria on the data set shown in Figure 2 using the above-detailed rules, specifically, those of the RDF type. The criteria is *predicate = CTO* and *object = Colin*. The depiction is simplified insofar as it shows execution of the rules serially: in practice, a preferred module 12 implemented in a rules engine (such as Jess) executes the rules in accord with the engine's underlying algorithm (e.g., a Rete algorithm as disclosed by Forgy, "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match," Problem Artificial Intelligence, 19(1982) 17-37, by http://herzberg.ca.sandia.gov/jess/docs/52/rete.html; or other underlying algorithm).

In a sequence of twelve frames, the depiction shows successive identification of triples as "related" (i.e., matching the criteria or related thereto) as each rule is applied or re-applied. The illustrated sequence proceeds from left-to-right then top-to-bottom, as indicated by the dashed-line arrows. For sake of simplicity, the data set is depicted in abstract in each frame, i.e., by a small directed graph of identical shape as that of Figure 2, but without the labels. Triples identified as related are indicated in black.

Referring to the first frame of Figure 3, the module 12 applies the Criteria Rule to the data set. Because the company://id#1—CTO—Colin triple matches the criteria (to repeat, *predicate = CTO* and *object = Colin),* it is identified as "related" and marked accordingly.

In the second frame, the module applies the Sibling Rule to find triples at the same level as the one(s) previously identified by the Criteria Rule. In this instance, the company://id#1—employee—Howard and company://id#1—employee—Alan triples are identified and marked accordingly.

In the third frame, the module applies the Ancestor Rule to walk up the directed graph to find ancestors of the triples previously identified as related. In this instance, the company://id#3—customer—company://id#1 triple is identified and marked accordingly.

In the fourth frame, the module applies the Descendent Rule to walk down the directed graph to find descendents of the triples previously identified as related. No triples are selected since both company://id#3—customer—company://id#2 and company://id#3—customer—company://id#4 share the same predicate as company://id#3—customer—company://id#1. Referring back to the detailed rules, company://id#2, by way of example, is a direct descendent that has a predicate (to wit, customer) connecting it with its identified direct ancestor (to wit, company://id#3) which matches a predicate that ancestor (to wit, company://id#3) has with a direct descendent (to wit, company://id#1) via which that direct ancestor (to wit, company://id#3) was identified during the execution of the Ancestor Rule.

In frames 5–8, the module 12 reapplies the rules, this time beginning with a Criteria Rule match of company://id#2—CTO—Colin. In frames 9–12, the module 12 finds no further matches upon reapplication of the rules.

Figure 4 parallels Figure 3, showing however application by module 12 of the criteria *predicate = employee* and *object = Alan* to the data set of Figure 2. Only eight frames are shown since module 12 finds no further matches during execution of the rules represented in the final four frames.

Of note in Figure 4 is frame two. Here, application of the Sibling Rule by module 12 does not result in identification of all of the siblings of company://id#1—employee—Alan (which had been identified as relevant in the prior execution of the Criteria Rule). This is because, one of siblings company://id#1—employee—Howard has the same predicate as that specified in the criteria. Accordingly, that triple is not identified or marked as related.

Figure 5 also parallels Figure 3, showing however application by module 12 of the criteria *resource = company://id#1* to the data set of Figure 2. Again, only eight frames are shown since module 12 finds no further matches during execution of the rules represented in the final four frames. Of note in Figure 5 is the identifications effected by specification of a *resource* as a criteria.

A further understanding of these examples may be attained by reference the Appendices A and B, filed herewith, which provide XML/RDF listings of the data sets and criteria, and which also show rule-by-rule identification or ("validation") of the triples.

Though the examples show application of the rules by module 12 to an RDF data set, it will be appreciated that alternate embodiments of the module can likewise apply the rules to data sets representing the meta directed graphs disclosed in copending, commonly assigned application United States Patent Application Number Serial No. 10/138,725, filed May 3, 2002, entitled "Methods And Apparatus for Visualizing Relationships Among Triples of Resource Description Framework (RDF) Data Sets," the teachings of which are incorporated herein by reference.

Described above are methods and apparatus meeting the desired objects. Those skilled in the art will, of course, appreciate that these are merely examples and that other embodiments, incorporating modifications to those described herein fall within the scope of the invention, of which we claim: